Network Working Group
Request for Comments: 4681
Updates: 4346
Category: Standards Track

S. Santesson
A. Medvinsky
J. Ball
Microsoft

October 2006

TLS User Mapping Extension

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document specifies a TLS extension that enables clients to send generic user mapping hints in a supplemental data handshake message defined in RFC 4680. One such mapping hint is defined in an informative section, the UpnDomainHint, which may be used by a server to locate a user in a directory database. Other mapping hints may be defined in other documents in the future.

Table of Contents

1. Introduction

This document has a normative part and an informative part. Sections 2-5 are normative. Section 6 is informative.

This specification defines a TLS extension and a payload for the SupplementalData handshake message, defined in RFC 4680 [N6], to accommodate mapping of users to their user accounts when using TLS client authentication as the authentication method.

The new TLS extension (user_mapping) is sent in the client hello message. Per convention defined in RFC 4366 [N4], the server places the same extension (user_mapping) in the server hello message, to inform the client that the server understands this extension. If the server does not understand the extension, it will respond with a server hello omitting this extension, and the client will proceed as normal, ignoring the extension, and not include the UserMappingDataList data in the TLS handshake.

If the new extension is understood, the client will inject UserMappingDataList data in the SupplementalData handshake message prior to the Client’s Certificate message. The server will then parse this message, extracting the client’s domain, and store it in the context for use when mapping the certificate to the user’s directory account.

No other modifications to the protocol are required. The messages are detailed in the following sections.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [N1].

The syntax for the TLS User Mapping extension is defined using the TLS Presentation Language, which is specified in Section 4 of [N2].

1.2. Design Considerations

The reason the mapping data itself is not placed in the extension portion of the client hello is to prevent broadcasting this information to servers that don’t understand the extension.

2. User Mapping Extension

A new extension type (user_mapping(6)) is added to the Extension used in both the client hello and server hello messages. The extension type is specified as follows.

```
   enum {
        user_mapping(6), (65535)
   } ExtensionType;
```

The "extension_data" field of this extension SHALL contain "UserMappingTypeList" with a list of supported hint types where:

```
   struct {
         UserMappingType user_mapping_types<1..2^8-1>;
   } UserMappingTypeList;
```

Enumeration of hint types (user_mapping_types) defined in this document is provided in Section 3.

The list of user_mapping_types included in a client hello SHALL signal the hint types supported by the client. The list of user_mapping_types included in the server hello SHALL signal the hint types preferred by the server.

If none of the hint types listed by the client is supported by the server, the server SHALL omit the user_mapping extension in the server hello.

When the user_mapping extension is included in the server hello, the list of hint types in "UserMappingTypeList" SHALL be either equal to, or a subset of, the list provided by the client.

3. User Mapping Handshake Exchange

The underlying structure of the SupplementalData handshake message, used to carry information defined in this section, is defined in RFC 4680 [N6].

A new SupplementalDataType [N6] is defined to accommodate communication of generic user mapping data. See RFC 2246 (TLS 1.0) [N2] and RFC 4346 (TLS 1.1) [N3] for other handshake types.

The information in this data type carries one or more unauthenticated hints, UserMappingDataList, inserted by the client side. Upon receipt and successful completion of the TLS handshake, the server

MAY use this hint to locate the user’s account from which user information and credentials MAY be retrieved to support authentication based on the client certificate.

```
   struct {
         SupplementalDataType supp_data_type;
         uint16 supp_data_length;
         select(SupplementalDataType) {
            case user_mapping_data: UserMappingDataList;
            }
   } SupplementalDataEntry;

   enum {
         user_mapping_data(0), (65535)
   } SupplementalDataType;
```

The user_mapping_data(0) enumeration results in a new supplemental data type UserMappingDataList with the following structure:

```
   enum {
         (255)
   } UserMappingType;

   struct {
          UserMappingType user_mapping_version;
          uint16 user_mapping_length;
          select(UserMappingType) { }
   } UserMappingData;

   struct{
      UserMappingData user_mapping_data_list<1..2^16-1>;
   }UserMappingDataList;
```

user_mapping_length
   This field is the length (in bytes) of the data selected by UserMappingType.

The UserMappingData structure contains a single mapping of type UserMappingType. This structure can be leveraged to define new types of user mapping hints in the future. The UserMappingDataList MAY carry multiple hints; it is defined as a vector of UserMappingData structures.

No preference is given to the order in which hints are specified in this vector. If the client sends more than one hint, then the Server SHOULD use the applicable mapping supported by the server.

Implementations MAY support the UPN domain hint as specified in Section 6 of this document. Implementations MAY also support other user mapping types as they are defined. Definitions of standards-track user mapping types must include a discussion of internationalization considerations.

4. Message Flow

In order to negotiate sending user mapping data to a server in accordance with this specification, clients MUST include an extension of type "user_mapping" in the (extended) client hello, which SHALL contain a list of supported hint types.

Servers that receive an extended client hello containing a "user_mapping" extension MAY indicate that they are willing to accept user mapping data by including an extension of type "user_mapping" in the (extended) server hello, which SHALL contain a list of preferred hint types.

After negotiation of the use of user mapping has been successfully completed (by exchanging hello messages including "user_mapping" extensions), clients MAY send a "SupplementalData" message containing the "UserMappingDataList" before the "Certificate" message. The message flow is illustrated in Figure 1 below.

```
      Client                                               Server

      ClientHello
       /* with user_mapping ext */ -------->
                                                      ServerHello
                                      /* with user-mapping ext */
                                                     Certificate*
                                               ServerKeyExchange*
                                              CertificateRequest*
                                   <--------      ServerHelloDone

      SupplementalData
       /* with UserMappingDataList */
      Certificate*
      ClientKeyExchange
      CertificateVerify*
      [ChangeCipherSpec]
      Finished                     -------->
                                               [ChangeCipherSpec]
                                   <--------             Finished
      Application Data             <------->     Application Data
```

* Indicates optional or situation-dependent messages that are not always sent according to RFC 2246 [N2] and RFC 4346 [N3].

Figure 1. Message Flow with User Mapping Data

The server MUST expect and gracefully handle the case where the client chooses not to send any supplementalData handshake message even after successful negotiation of extensions. The client MAY at its own discretion decide that the user mapping hint it initially intended to send no longer is relevant for this session. One such reason could be that the server certificate fails to meet certain requirements.

5. Security Considerations

The user mapping hint sent in the UserMappingDataList is unauthenticated data that MUST NOT be treated as a trusted identifier. Authentication of the user represented by that user mapping hint MUST rely solely on validation of the client certificate. One way to do this is to use the user mapping hint to locate and extract a certificate of the claimed user from the trusted directory and subsequently match this certificate against the validated client certificate from the TLS handshake.

As the client is the initiator of this TLS extension, it needs to determine when it is appropriate to send the User Mapping Information. It may not be prudent to broadcast a user mapping hint to just any server at any time.

To avoid superfluously sending user mapping hints, clients SHOULD only send this information if it recognizes the server as a legitimate recipient. Recognition of the server can be done in many ways. One way to do this could be to recognize the name and address of the server.

In some cases, the user mapping hint may itself be regarded as sensitive. In such cases, the double handshake technique described in [N6] can be used to provide protection for the user mapping hint information.

6. UPN Domain Hint (Informative)

This specification provides an informative description of one user mapping hint type for Domain Name hints and User Principal Name hints. Other hint types may be defined in other documents in the future.

The User Principal Name (UPN) in this hint type represents a name that specifies a user’s entry in a directory in the form userName@domainName. Traditionally, Microsoft has relied on the presence of such a name form to be present in the client certificate when logging on to a domain account. However, this has several drawbacks since it prevents the use of certificates with an absent UPN and also requires re-issuance of certificates or issuance of multiple certificates to reflect account changes or creation of new accounts. The TLS extension, in combination with the defined hint type, provides a significant improvement to this situation as it allows a single certificate to be mapped to one or more accounts of the user and does not require the certificate to contain a proprietary UPN.

The domain_name field MAY be used when only domain information is needed, e.g., where a user have accounts in multiple domains using the same username name, where that user name is known from another source (e.g., from the client certificate). When the user name is also needed, the user_principal_name field MAY be used to indicate both username and domain name. If both fields are present, then the server can make use of whichever one it chooses.

```
enum {
       upn_domain_hint(64), (255)
} UserMappingType;
```

```
      struct {
             opaque user_principal_name<0..2^16-1>;
             opaque domain_name<0..2^16-1>;
      } UpnDomainHint;

      struct {
             UserMappingType user_mapping_version;
             uint16 user_mapping_length;
             select(UserMappingType) {
                   case upn_domain_hint: UpnDomainHint;
             }
      } UserMappingData;
```

The user_principal_name field, when specified, SHALL be of the form "user@domain", where "user" is a UTF-8 encoded Unicode string that does not contain the "@" character, and "domain" is a domain name meeting the requirements in the following paragraph.

The domain_name field, when specified, SHALL contain a domain name [N5] in the usual text form; in other words, a sequence of one or more domain labels separated by ".", each domain label starting and ending with an alphanumeric character and possibly also containing "-" characters. This field is an "IDN-unaware domain name slot" as defined in RFC 3490 [N7], and therefore, domain names containing non-ASCII characters have to be processed as described in RFC 3490 before being stored in this field.

The UpnDomainHint MUST at least contain a non-empty user_principal_name or a non-empty domain_name. The UpnDomainHint MAY contain both user_principal_name and domain_name.

7. IANA Considerations

IANA has taken the following actions:

1) Created an entry, user_mapping(6), in the existing registry for ExtensionType (defined in RFC 4366 [N4]).

2) Created an entry, user_mapping_data(0), in the new registry for SupplementalDataType (defined in RFC 4680).

3) Established a registry for TLS UserMappingType values. The first entry in the registry is upn_domain_hint(64). TLS UserMappingType values in the inclusive range 0-63 (decimal) are assigned via RFC 2434 [N8] Standards Action. Values from the inclusive range 64-223 (decimal) are assigned via RFC 2434 Specification Required. Values from the inclusive range 224-255 (decimal) are reserved for RFC 2434 Private Use.

8. Normative References

[N1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[N2] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[N3] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[N4] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", RFC 4366, April 2006.

[N5] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

[N6] Santesson, S., "TLS Handshake Message for Supplemental Data", RFC 4680, October 2006.

[N7] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[N8] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

9. Acknowledgements

The authors extend a special thanks to Russ Housley, Eric Resocorla, and Paul Leach for their substantial contributions.

Authors' Addresses

Stefan Santesson
Microsoft
Finlandsgatan 30
164 93 KISTA
Sweden

EMail: stefans@microsoft.com

Ari Medvinsky
Microsoft
One Microsoft Way
Redmond, WA 98052-6399
USA

EMail: arimed@microsoft.com

Joshua Ball
Microsoft
One Microsoft Way
Redmond, WA 98052-6399
USA

EMail: joshball@microsoft.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).